

CLAIM AMENDMENTS

1 1.-15. (Canceled)

1 16. (Currently Amended) A method for routing or switching data packets,
2 ~~including~~comprising the computer-implemented steps of:
3 receiving a data packet at an input interface on a router or switch;
4 looking up information in the header of said data packet in an expanded M-trie data
5 structure, wherein said expanded M-trie data structure is organized as a multi-level
6 tree including a root node, inferior nodes, and terminal nodes, wherein each node
7 includes an address and an opcode; and
8 terminating said ~~lookup~~step of looking up information; ~~and~~
9 ~~routing said data packet at one or more output interfaces on said router or said switch.~~

1 17. (Canceled)

1 18. (Previously Presented) A method as in claim 16, wherein said opcode describes an
2 operation to be performed that is based upon data included in a packet header, so as to
3 facilitate a lookup of said packet header.

1 19. (Previously Presented) A method as in claim 16, wherein said address includes the
2 address of a node in said expanded M-trie data structure that is to be traversed.

1 20. (Original) A method as in claim 16, wherein said expanded M-trie data structure
2 includes a set of access control parameters.

1 21. (Previously Presented) A method as in claim 16, wherein said expanded M-trie data
2 structure includes a set of Quality of Service (QoS) parameters.

1 22. (Previously Presented) A method as in claim 16, wherein said expanded M-trie data
2 structure includes a set of Class of Service (CoS) parameters.

1 23. (Previously Presented) A method as in claim 16, wherein said nodes include opcodes for
2 demultiplexing, opcodes for matching, and opcodes for hashing.

1 24. (Previously Presented) A method as in claim 23, wherein said opcodes for
2 demultiplexing include instructions to demultiplex into branches of said expanded M-trie
3 data structure based on contents of a byte of said packet header that is being read.

1 25. (Previously Presented) A method as in claim 23, wherein said opcodes for matching
2 include instructions to compare the contents of a given byte of the flow label to given node
3 data.

1 26. (Previously Presented) A method as in claim 23, wherein said opcodes for hashing
2 include instructions to hash into different M-trie plus branches based on the contents of a
3 given byte in said packet header.

1 27. (Canceled)

1 28. (Previously Presented) An apparatus for routing or switching data packets, comprising a
2 device that performs a method comprising:
3 storing in memory an M-trie data structure, said data structure organized as a multi-level tree
4 having a set of nodes, including a root node, inferior nodes and terminal nodes,
5 wherein each node includes an address and an opcode;
6 receiving a data packet at an input interface on a router or switch, wherein the data packet
7 includes information in at least a header with at least a field that is used by said M-
8 trie data structure to indicate an action for said device to perform in order to select a
9 leaf associated with said M-trie data structure;
10 looking up the information, wherein the looking up includes performing the action; and

11 routing said data packet at one or more output interfaces on said router or said switch.

1 29. (Currently Amended) A method for routing or switching data packets, comprising the
2 computer-implemented steps of:
3 storing in memory an M-trie data structure, said data structure organized as a multi-level tree
4 having a set of nodes, including a root node, inferior nodes and terminal nodes,
5 wherein each node includes an address and an opcode;
6 receiving a data packet at an input interface on a router or switch, wherein the data packet
7 includes information in at least a header with at least a field that is used by said M-
8 trie data structure to indicate an action for a router to perform in order to select a leaf
9 associated with said M-trie data structure; and
10 looking up the information, wherein the looking up includes performing the action; ~~and~~
11 ~~routing said data packet at one or more output interfaces on said router or said switch.~~

1 30. (Previously Presented) A memory storing a program for performing a method for
2 routing or switching data packets, comprising:
3 storing in memory an M-trie data structure, said data structure organized as a multi-level tree
4 having a set of nodes, including a root node, inferior nodes and terminal nodes,
5 wherein each node includes an address and an opcode;
6 receiving a data packet at an input interface on a router or switch, wherein the data packet
7 includes information in at least a header with at least a field that is used by said M-
8 trie data structure to indicate an action for a router to perform in order to select a leaf
9 associated with said M-trie data structure;
10 looking up the information, wherein the looking up includes performing the action; and
11 routing said data packet at one or more output interfaces on said router or said switch.

1 31. (Canceled)

1 32. (Previously Presented) A memory as in claim 30, wherein said address includes an
2 address of a node in said M-trie data structure that is to be traversed.

- 1 33. (Previously Presented) A memory as in claim 30, wherein said M-trie data structure
2 includes a set of access control parameters.
- 1 34. (Previously Presented) A memory as in claim 30, wherein said M-trie data structure
2 includes a set of Quality of Service (QoS) parameters.
- 1 35. (Previously Presented) A memory as in claim 30, wherein said expanded M-trie data
2 structure includes a set of Class of Service (CoS) parameters.
- 1 36. (Previously Presented) A memory as in claim 30 wherein at least one of the root node,
2 inferior nodes, or the terminal node includes an opcode for demultiplexing, an
3 opcode for matching, and an opcode for hashing.
- 1 37. (Previously Presented) A memory as in claim 36 wherein said opcode for
2 demultiplexing includes instructions to demultiplex into branches of the M-trie data
3 structure based on contents of a byte of said packet header.
- 1 38. (Previously Presented) A method as in claim 36, wherein said opcode for matching
2 includes instructions to compare the contents of a given byte of a flow label to given
3 node data.
- 1 39. (Previously Presented) A method as in claim 36, wherein said opcode for hashing
2 includes instructions to hash into different branches the M-trie data structure based
3 on the contents of a given set of bytes in said packet header.
- 1 40. (New) A method as recited in Claim 16, further comprising routing said data packet
2 at one or more output interfaces on said router or said switch.
- 1 41. (New) A method as recited in Claim 16, further comprising determining, based on
2 one or more Access Control List (ACL) criteria stored in said expanded M-trie data
3 structure, whether to drop or forward said data packet.

- 1 42. (New) A method as recited in Claim 41, wherein determining whether to drop or
2 forward said data packet comprises matching said information in the header of said
3 data packet to the one or more ACL criteria stored in said expanded M-trie data
4 structure.
- 1 43. (New) A method as recited in Claim 41, wherein said one or more ACL criteria
2 include at least one of a source address, destination address, and upper-layer protocol
3 information.
- 1 44. (New) A method as recited in Claim 41, wherein said one or more ACL criteria are
2 stored in a sub-tree of said expanded M-trie data structure.
- 1 45. (New) A method as recited in Claim 29, further comprising routing said data packet
2 at one or more output interfaces on said router or said switch.
- 1 46. (New) A method as recited in Claim 29, further comprising determining, based on
2 one or more Access Control List (ACL) criteria stored in said M-trie data structure,
3 whether to drop or forward said data packet.
- 1 47. (New) A method as recited in Claim 46, wherein determining whether to drop or
2 forward said data packet comprises matching said information to the one or more
3 ACL criteria stored in said M-trie data structure.
- 1 48. (New) A method as recited in Claim 46, wherein said one or more ACL criteria
2 include at least one of a source address, a destination address, and upper-layer
3 protocol information.
- 1 49. (New) A method as recited in Claim 46, wherein said one or more ACL criteria are
2 stored in a sub-tree of said M-trie data structure.